PCT/EP2004/012226

PATENTS • DESIGNS The Patent Office COPYRIGHT • TRADE MARKS

INVESTOR IN PEOPLE

EPO - DG 1

13. 12. 2004

94

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed R. Mahoney

Dated 9 November 2004

Patents Form 1/77

Patents Act 1977
(Rule 16)

THE PATENT OFFICE
CR
29 OCT 2003
RECEIVED BY FAX

The Patent Office

290CT03 E848106-1 D00393
P01/7700 0.00-0325225.1

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

29 OCT 2003

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference	P/64001.GBA
2. Patent application number *(The Patent Office will fill in this part)*	0325225.1
3. Full name, address and postcode of the or of each applicant *(underline all surnames)*	Argelcom Limited Osborne Clarke 2 Temple Back East Temple Quay Bristol, BS1 6EG
Patents ADP number *(if you know it)*	08743171001
If the applicant is a corporate body, give the country/state of its incorporation	United Kingdom
4. Title of the invention	A secure communication method
5. Name of your agent *(if you have one)*	N G McGowan
"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*	Intellectual Property Department Siemens Shared Services Siemens House Oldbury, Bracknell Berkshire, RG12 8FZ United Kingdom
Patents ADP number *(if you know it)*	07761000002

	Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*
6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number			

	Number of earlier application	Date of filing *(day / month / year)*
7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application		

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

Patents Form 1/77

0084180 29-Oct-03 02:33

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	-
Description	20
Claim *(s)*	-
Abstract	-
Drawing *(s)*	-

[illegible]

10. If you are also filing any of the following, state how many against each item.

Priority documents	-
Translations of priority documents	-
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	-
Request for preliminary examination and search *(Patents Form 9/77)*	-
Request for substantive examination *(Patents Form 10/77)*	-
Any other documents *(please specify)*	-

11. I/We request the grant of a patent on the basis of this application.

Signature [illegible] Date 29 Oct 2003

N G McGowan

12. Name and daytime telephone number of person to contact in the United Kingdom N G McGowan - 01344 396808

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

Patents Form 1/77

0084180 29-Oct-03 02:33

A secure communication method

In summary the main invention is related to a secure email solution.

1. The concept of an m-KEM. This is a method which takes as input a set of public keys and then produces a session key plus an encapsulation (encryption) of that session key under all the public keys. The method comes with an associated decapsulation method which allows any body who knows the secret key associated with any of the public keys to recover the session key from the encapsulation.
    a. An instantiation of the above by naively concatenating encryptions of the session key together.
    b. An efficient instantiation of the above using a variant of the ElGamal encryption scheme.
2. The concept of an m-ID encryption scheme. This takes as input a set of identities and allows the encryption of an arbitrary message to one of these identities, such that any user with the associated private key for any of the identities is able to decrypt the message.
    a. An efficient instantiation of an m-ID encryption scheme using an adaption of the Boneh-Franklin encryption scheme based on pairings on elliptic curves.
3. The concept of an ID-m-KEM. This is like the above m-KEM except that now the public keys are replaced with simple identity strings. Thus allowing the use of an m-KEM in the identity based setting as well as the public key setting.
    a. An efficient instantiation of an ID-m-KEM using the m-ID encryption scheme.
4. The concept of using the transform $s = s+1/s$ so as to remove the need for transmitting y-coordinates or using point compression in schemes based on pairings. Such schemes would include the Boneh-Franklin ID-based encryption scheme, the BLS signature scheme, the Hess ID-based signature scheme plus the m-ID and ID-m-KEM mentioned above.
5. The concept of adding trust authority keys together so as to create intersecting domains of trust in an ID-based infrastructure.

Clearly the above ideas can be applied to any system which wishes to enable secure messeges being sent and is not just restricted to the email environment.

m-KEM's and ID-m-KEM's

Abstract. We present the notion of an m-KEM, which is a Key Encapsulation Mechanism (KEM) which takes multiple public keys as input. This has applications where one wishes to encrypt a single large document to a set of multiple recipients, as when one sends an encrypted email to more than one person. We present a security model and show that the naive approach to defining an m-KEM is secure in this model. We then go on to present a more efficient construction of an m-KEM. Finally we turn to identity based variants of an m-KEM and give a construction, based on pairings, which is significantly more efficient than the naive application of the Boneh–Franklin scheme applied a multiple number of times.

1 Introduction

Public key cryptography has been traditionally concerned with two parties communicating. In the traditional scenario one party, Alice, wishes to communicate securely with one other party, Bob. Alice obtains Bob's authentic public key and then encrypts the data she wishes to send to Bob. Bob, knowing the associated private key, is able to decrypt the ciphertext to obtain Alice's message. Since public key algorithms are very slow, if Alice wishes to send a large amount of data she first encrypts a per message symmetric "session key" to Bob using Bob's public key algorithm and then encrypts the actual data using a fast symmetric cipher keyed by the session key. Such a combination of public key and symmetric techniques is called a hybrid encryption algorithm.

This hybrid technique has been strengthened in recent years with the use of the KEM-DEM philosophy, see [6] and [7]. In this approach to hybrid encryption one defines a symmetric data encapsulation mechanism (DEM) which takes a key K and a message M and computes

$$C \leftarrow DEM_K(M).$$

Given knowledge of K one can also recover M via

$$M \leftarrow DEM_K^{-1}(C).$$

To transport the key K to the recipient of the ciphertext the sender uses a key encapsulation mechanism (KEM). This is an algorithm which takes as input a public key pk and outputs a session key K plus an encapsulation E of this session key under the public key,

$$(K, E) \leftarrow KEM(\text{pk}).$$

Notice, that the session key is not used as input to the KEM. The recipient recovers the key K using his private key sk via the decapsulation mechanism

$$K \leftarrow KEM^{-1}(E, \text{sk}).$$

The full ciphertext of the message M is then given by

$$E \| C.$$

The use of the KEM-DEM philosophy allows the different components of a hybrid encryption scheme to be designed in isolation, leading to a simpler analysis and hopefully more efficient schemes.

However, as soon as one moves away from the traditional two-party setting problems occur. Suppose Alice now wishes to send a large file to two parties (say Bob and Charlie), for example she may wish to encrypt an email to Bob and Charlie, or encrypt a file on her system such that either Bob or Charlie can decrypt it. From ones own experience one notices that very few emails are sent to a single recipient, hence such a one-to-many model is clearly of importance.

A number of possible solutions exist to this problem. All of which have disadvantages. In the first naive solution one simply encrypts the data twice, once for Bob and once for Charlie, using their respective public key schemes. This is clearly wasteful especially if the data to be encrypted is large. A more efficient solution would be to encrypt the data once with a symmetric encryption key K and then encrypt this key under Bob and Charlie's public keys, i.e. the ciphertext would look like

$$\mathcal{E}_{\text{pk}_B}(K) \| \mathcal{E}_{\text{pk}_C}(K) \| \mathcal{E}_K(M).$$

Whilst this is probably sufficient for two users, this can become very expensive for a large number of users.

In addition it is unclear what security model one is using for such a scheme. The work of Bellare, Boldyreva and Micali [2] looks at the security of encryption schemes in the presence of many users but did not consider the fact that a "ciphertext" could correspond to different users. In their model the above hybrid encryption to two parties would be considered as two encryptions, whereas we wish to treat it as a single encryption.

The use of the KEM-DEM philosophy in such a situation is also not applicable. After all the KEM produces the session key, hence application of a KEM for two users would result in two different session keys. What is required is a KEM like construction which takes as input n public keys and outputs a session key and an encapsulation of that session key under each of the input public keys. We would like such a multiple KEM (or m-KEM) which is more efficient than the above concatenation of public key encryptions of a session key.

In this paper we propose such m-KEMs and propose a security model that they should possess. We show that the above naive concaternation method is secure in this model, although inefficient. We then present a public key m-KEM based on the Diffie–Hellman problem which is more efficient than repeated encryption of a session key using an analogous traditional public key system.

In addition we present ID based variants of our ideas. The use of identity based encryption has become of interest since the ground breaking work of Boneh and Franklin [3]. We present an ID based m-KEM which is much more efficient for encrypting to a large number of identities than repeated use of the Boneh–Franklin scheme for encrypting a session key. Our construction makes use of the Boneh–Franklin scheme but also uses a special case of the construction of Smart [8] for encrypting to arbitrary policies.

2 Notation

We let $v \leftarrow u$ for variables v and u to denote assignment. For a set S we let $v \leftarrow S$ denote the variable v being assigned the set S and $v \overset{R}{\leftarrow} S$ to denote v being assigned an element of the set S chosen uniformly at random.

If A is a, possible probabilistic, algorithm then $v \leftarrow A$ denotes v being assigned the output of algorithm A with the probability distribution induced by A's input and internal random choices. If we wish to make explicit precisely what value r is used as the randomness in a probabilistic algorithm $A(x)$ with input x we write $A(x;r)$.

A function f is said to be negligible if for all polynomials p there exists a constant N_p such that $f(x) \leq \frac{1}{p(x)}$ for all $x \geq N_p$.

3 Security of a KEM

A KEM (key encapsulation mechanism) is a public key scheme which allows a sender to generate an encryption of a random session key, and allows the holder of the correct private key to recover the session key from the ciphertext. We let $\mathbb{D}$ denote a set of domain parameters which could consist of only the security parameter written in unary 1^k, or could consist of a public group and generator as in ElGamal systems.

More formally we define a KEM is a triple of algorithms:

- $\mathcal{G}_{KEM}(\mathbb{D})$ which is a probabilistic key generation algorithm. On input of $\mathbb{D}$ this algorithm outputs a public/private key pair (pk, sk).
- $\mathcal{E}_{KEM}(\text{pk})$ which is a probabilistic encapsulation algorithm. On input of a public key pk this algorithm outputs an encapsulated key-pair (K, C), where $K \in \mathbb{K}$ is the session key and C is an encapsulation of the key K under the public key pk. In other words C is a ciphertext of the message K. We assume that the space $\mathbb{K}$ of all keys output by $\mathcal{E}$ are of some fixed length.

- $\mathcal{D}_{KEM}(C, \mathrm{sk})$ which is a decapsulation algorithm. This takes as input an encapsulation C and a private key sk and outputs a key K or a special symbol $\perp$ representing the case where C is an invalid encapsulation with respect to the private key sk.

For such a scheme to be useful we require that it is sound in the following sense,

$$\Pr\left((\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{G}_{KEM}(\mathbb{D}), (K, C) \leftarrow \mathcal{E}_{KEM}(\mathrm{pk}) : K = \mathcal{D}_{KEM}(C, \mathrm{sk})\right) = 1.$$

Security of a KEM is defined in the following way. We assume an adversary $\mathcal{A}$ which runs in two stages. In the first stage it is allowed to produce encapsulations and (depending on the precise security definition we require) it may be allowed access to a decapsulation oracle on encapsulations of its choosing. At the end of this stage it returns some state information.

In the second stage it is provided with a challenge encapsulation C^*, its state information from the first stage plus two keys K_0 and K_1. The adversaries goal in the second stage is to decide which key K_b is encapsulated by C. In this second stage it may also have access to an decapsulation oracle, but if it does it is not allowed to request the decapsulation of the challenge C^*.

Consider the following game played with such an adversary:

```
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{G}_{KEM}(\mathbb{D})$.
$s \leftarrow \mathcal{A}^1(\mathrm{pk})$.
$b \xleftarrow{R} \{0, 1\}$.
$(K_b, C^*) \leftarrow \mathcal{E}_{KEM}(\mathrm{pk})$.
$K_{1-b} \xleftarrow{R} \mathbb{K}$.
$b' \leftarrow \mathcal{A}^2(C^*, \{K_0, K_1\}, s)$.
Output whether $b = b'$.
```

The adversary is said to win the game if $b = b'$. The advantage of an adversary is defined to be

$$\mathrm{Adv}_{\mathcal{A}} = |\Pr(b = b') - 1/2|.$$

If the maximum advantage over all possible adversaries $\mathcal{A}$ is a negligible function of the security parameter k then we say that the KEM is IND-xxx secure, where xxx denotes what access $\mathcal{A}$ is allowed to a decapsulation oracle. If $\mathcal{A}$ is not allowed any access to such an oracle then we say the scheme is IND-CPA secure, if it is only allowed access during stage one then we say the scheme is IND-CCA1 secure and if it is allowed access in both stages (subject to the earlier restriction on requesting the decapsulation of C^*) then we say the scheme is IND-CCA2 secure.

A KEM needs to be used with a DEM (data encapsulation mechanism) to provide a hybrid encryption algorithm. A DEM is a symmetric encryption algorithm which takes a symmetric key k and a message (resp. ciphertext) and provides the corresponding ciphertext (resp. message). Security definitions can be provided for such DEMs, which are independent of the security definition of the associated KEM. The combined KEM-DEM encryption scheme is said to be secure in the standard IND-CCA2 sense (for a public key encryption scheme) if the DEM is secure and the KEM is secure in the IND-CCA2 sense above. Hence, the goal is to define KEM's which are IND-CCA2 secure.

4 m-KEMs and ID-m-KEMs

We now extend the notion of KEM to deal with the case where one wants to encrypt a large amount of data to multiple people, say n people. In such a situation it makes sense to apply the DEM once and so one requires a mechanism which creates the symmetric key for the DEM and an encapsulation which encapsulates the key to many parties at once. We call such a system an m-KEM for "multiple KEM".

We also introduce the notion of an ID-KEM and an ID-m-KEM which are the analogous definitions in an ID-based setting, as opposed to a traditional public key setting. In the following sections we define these concepts and provide appropriate security definitions.

4.1 m-KEMs

Note, a trivial solution would be to generate a session key K for the DEM and then encrypt this to the various intended receivers by encrypting using an IND-CCA2 public key encryption algorithm. This would produce n distinct ciphertexts $c_1, \dots, c_n$, each encrypting K for n different users. We would then define the key encapsulation as

$$C = c_1 \| c_2 \| \cdots \| c_n.$$

Our goal however is to do this in a more efficient manner, where one measures efficiency either in terms of computing resources or in terms of length of the resulting encapsulation C.

Note, in the above trivial system one would need to specify which ciphertext component c_i corresponded to which user u_i. Hence, the ciphertext should actually contain some information specifying which ciphertext corresponds to which user, i.e. we need to have something like

$$C = u_1 \| c_1 \| u_2 \| c_2 \cdots \| u_n \| c_n.$$

In our future discussion we shall drop this explicit reference to which users which component corresponds to. Instead we shall pass the list of recipients to the decryption function as an optional additional parameter.

Just as for a KEM, we define an m-KEM formally as a triple of algorithms, $(\mathcal{G}_{mKEM}, \mathcal{E}_{mKEM}, \mathcal{D}_{mKEM})$, by adapting the earlier definition, we have

- $\mathcal{G}_{mKEM}(\mathbb{D})$ which is a probabilistic key generation algorithm. On input of $\mathbb{D}$, the domain parameters, this algorithm outputs a public/private key pair $(\mathrm{pk}, \mathrm{sk})$.

- $\mathcal{E}_{mKEM}(\mathcal{P})$ which is a probabilistic encapsulation algorithm. On input of a set of public key $\mathcal{P} = \{\text{pk}_1, \ldots, \text{pk}_n)$ this algorithm outputs an encapsulated key-pair (K, C), where $K \in \mathbb{K}$ is the session key and C is an encapsulation of the key K under the public keys $\{\text{pk}_1, \ldots, \text{pk}_n)$.
- $\mathcal{D}_{mKEM}(C, sk, \mathcal{P})$ which is a decapsulation algorithm. This takes as input an encapsulation C and a private key sk, plus optionally the set of all recipients $\mathcal{P}$, and outputs a key K or a special symbol $\perp$ representing the case where C is an invalid encapsulation with respect to the private key sk.

Soundness is now defined as follows.

$$\Pr\left(\begin{array}{ll}(\text{pk}_i, \text{sk}_i) \leftarrow \mathcal{G}_{mKEM}(\mathbb{D}) \forall i \in \{1, \ldots, n\} & \\ (K, C) \leftarrow \mathcal{E}_{mKEM}(\{\text{pk}_1, \ldots, \text{pk}_n\}), & \\ j \xleftarrow{R} \{1, \ldots, n\} & : K = \mathcal{D}_{mKEM}(C, \text{sk}_j)\end{array}\right) = 1.$$

Security of an m-KEM is defined in a similar manner to a KEM via the following game.

$(\text{pk}_i, \text{sk}_i) \leftarrow \mathcal{G}_{mKEM}(\mathbb{D}) \forall i \in \{1, \ldots, n\}$.
$\mathcal{P}' \leftarrow \{\text{pk}_1, \ldots, \text{pk}_n\}$
$\{s, \mathcal{P}\} \leftarrow A^1(\mathcal{P}')$, where $\mathcal{P} \subset \mathcal{P}'$ and $m = \#\mathcal{P} \leq n$.
$b \xleftarrow{R} \{0, 1\}$.
$(K_b, C^*) \leftarrow \mathcal{E}_{mKEM}(\mathcal{P})$.
$K_{1-b} \xleftarrow{R} \mathbb{K}$.
$b' \leftarrow A^2(C^*, \{K_0, K_1\}, s)$.
Output whether $b = b'$.

Notice in stage one the adversary picks a set $\mathcal{P}$ of public keys on which it wants to be challenged on. One needs to be careful about the oracle access to the decapsulation oracle. To see why consider our earlier trivial mKEM with two parties, the challenge is given by

$$C^* = c_1 \| c_2.$$

However, using a traditional CCA2 definition of security an adversary could produce the encapsulation

$$C = c_1$$

and ask the decapsulation oracle to return the associated private key. Since $C \neq C^*$ this is a valid oracle query, which would result in the adversary breaking the system.

However, we feel such an oracle query is too stringent for our purposes. We therefore restrict decapsulation oracle queries in the second stage to be only allowed if the resulting key is different from the key encapsulated by C^*. Such a restricted oracle access is used in other works to deal with the public key encryption algorithms which suffer from benign malleability.

We say an m-KEM is (m,n)-IND-CCA2 secure, for an integers n and m with $m \leq n$, if the advantage of the adversary winning the above game is negligible as a function of k. We assume the adversary is allowed access to decapsulation oracle queries in both stages, subject to the above restriction on C^*.

4.2 ID-m-KEMs

We now turn to providing a definition for an ID-based m-KEM. A definition for an ID based equivalent of a KEM (i.e. with respect to a single identity) follows from the general definition by taking the number of identities equal to one. Hence we shall not discuss ID-KEMs further and shall concentrate on the more general ID-m-KEMs.

An ID based cryptosystem, such as that of Boneh and Franklin [3] or Cocks [4], is based on a trust authority who generates a single public/private key pair at setup for themselves. Users public keys are then given by a deterministic function of their identities, the associated private key they obtain from the trust authority by means of a so-called extraction query.

We therefore define a identity based multiple KEM, or ID-m-KEM, formally as a quartet of algorithms $(\mathcal{G}, \mathcal{X}, \mathcal{E}, \mathcal{D})$ as before where, by adapting the earlier definitions, we have

- $\mathcal{G}_{ID-mKEM}(\mathbb{D})$ which is a probabilistic key generation algorithm for the trust authority. On input of $\mathbb{D}$, the domain parameters, this algorithm outputs a public/private key pair (pk, sk).
- $\mathcal{X}_{ID-mKEM}(\mathrm{ID}, \mathrm{sk})$ is the trust authorities key extraction algorithm. It takes as input the trust authorities private key sk and an identity string ID and outputs the associated secret key S_{ID}.
- $\mathcal{E}_{ID-mKEM}(\mathcal{I}, \mathrm{pk})$ which is a probabilistic encapsulation algorithm. On input of a set of identities $\mathcal{I} = \{\mathrm{ID}_1, \ldots, \mathrm{ID}_n)$ and the trust authorities public key pk, this algorithm outputs an encapsulated key-pair (K, C), where $K \in \mathbb{K}$ is the session key and C is an encapsulation of the key K under the identities $\{\mathrm{ID}_1, \ldots, \mathrm{ID}_n\}$.
- $\mathcal{D}_{ID-mKEM}(C, S_{\mathrm{ID}}, \mathrm{pk}, \mathcal{I})$ which is a decapsulation algorithm. This takes as input an encapsulation C, a private key S_{ID} and the trust authority public key pk, plus optionally the set of all recipients $\mathcal{I}$. It then outputs a key K or a special symbol $\perp$ representing the case where C is an invalid encapsulation with respect to the private key S_{ID}.

Soundness is now defined as follows.

$$\Pr\left(\begin{array}{l}(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{G}_{ID-mKEM}(\mathbb{D}),\\ S_{\mathrm{ID}_i} \leftarrow \mathcal{X}_{ID-mKEM}(\mathrm{ID}_i, \mathrm{sk}) \forall i \in \{1, \ldots, n\},\\ (K, C) \leftarrow \mathcal{E}_{ID-mKEM}(\{\mathrm{ID}_1, \ldots, \mathrm{ID}_n\}, \mathrm{pk}),\\ j \xleftarrow{R} \{1, \ldots, n\}\\ \qquad\qquad : K = \mathcal{D}_{ID-mKEM}(C, S_{\mathrm{ID}_j}, \mathrm{pk})\end{array}\right) = 1.$$

Security is now defined with respect to the following game:

$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{G}_{ID-mKEM}(\mathbb{D})$.
$S_{\mathrm{ID}_i} \leftarrow \mathcal{X}_{ID-mKEM}(\mathrm{ID}_i, \mathrm{sk}) \forall i \in \{1, \ldots, n\}$.
$\mathcal{P}' \leftarrow \{\mathrm{ID}_1, \ldots, \mathrm{ID}_n\}$
$\{s, \mathcal{P}\} \leftarrow \mathcal{A}^1(\mathcal{P}')$, where $\mathcal{P} \subset \mathcal{P}'$ and $m = \#\mathcal{P} \leq n$.

```
$b \xleftarrow{R} \{0,1\}$.
$(K_b, C^*) \leftarrow \mathcal{E}_{ID-mKEM}(\mathcal{P}, \mathrm{pk})$.
$K_{1-b} \xleftarrow{R} \mathbb{K}$.
$b' \leftarrow \mathcal{A}^2(C^*, \{K_0, K_1\}, s)$.
Output whether $b = b'$.
```

Except now, for full (m, n)-ID-IND-CCA2 security, we allow the adversary $\mathcal{A}$ not only access to a decapsulation oracle, we also allow access to a secret key extraction oracle that will output the key S_{ID} for any queried identity ID. These oracles accesses are subject to the following provisos

- In the second stage it cannot ask for the decapsulation of any encapsulation C which would result in the same key K as that encapsulated by C^*. Hence, we allow benign forms of malleability.
- If ID is queried to $\mathcal{X}_{ID-mKEM}$ in the first stage then ID $\notin \mathcal{P}$.
- If ID $\in \mathcal{P}$ then the oracle $\mathcal{X}_{ID-mKEM}$ may not be queried with ID in the second stage.

5 Constructions of m-KEMs

We start this section by giving a generic construction which mirrors the naive construction of the introduction. Then we go on to provide a more efficient construction based on the ElGamal encryption function.

5.1 A Generic Construction

We let $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ denote a public key encryption algorithm which is IND-CCA2 secure. We define a KEM from $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ as follows, where $\mathcal{M}$ is the message space and $\mathcal{R}$ is the space of randomness used by $\mathcal{E}$,

```
$\mathcal{G}_{mKEM}(\mathbb{D})$:
    $(\mathrm{sk}, \mathrm{pk}) \leftarrow \mathcal{G}(\mathbb{D})$.
    Output $(\mathrm{pk}, \mathrm{sk})$.

$\mathcal{E}_{mKEM}(\{\mathrm{pk}_1, \ldots, \mathrm{pk}_n\})$:
    $m \xleftarrow{R} \mathcal{M}$.
    $r_i \xleftarrow{R} \mathcal{R}$ for all $i$.
    $c_i \leftarrow \mathcal{E}(m, \mathrm{pk}_i; r_i)$ for all $i$.
    $K \leftarrow KDF(m)$.
    $C \leftarrow (c_1, \ldots, c_n)$.
    Output $(K, C)$.

$\mathcal{D}_{mKEM}(C, \mathrm{sk}_i)$:
    Parse $C$ as $(c_1, \ldots, c_n)$.
    $m \leftarrow \mathcal{D}(c_i, \mathrm{sk}_i)$.
    If $m = \perp$ then output $\perp$ and halt.
    $K \leftarrow KDF(m)$.
    Output $K$.
```

Later we shall show

Theorem 1. *If $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is IND-CCA2 secure as a public key encryption scheme and n grows as a polynomial function of the security parameter then the m-KEM $(\mathcal{G}_{mKEM}, \mathcal{E}_{mKEM}, \mathcal{D}_{mKEM})$ is (m, n)-IND-CCA2 for all $m \leq n$.*

5.2 An efficient ElGamal based m-KEM

We now present an efficient m-KEM based on the ElGamal encryption algorithm for a group G of prime order $q \approx 2^k$ with generator g. We let $\mathbb{D} = \{q, g, G\}$ denote the domain parameters of the scheme, ElGamal is then given by the triple of algorithms:

```
G(D):                              E(m, pk; r):               D((c_1, c_2), sk):
    $\mathrm{sk} \xleftarrow{R} \mathbb{F}_q^*$.          $c_1 \leftarrow g^r$.            $m = c_2/c_1^{\mathrm{sk}}$.
    $\mathrm{pk} \leftarrow g^{\mathrm{sk}}$.             $c_2 \leftarrow m \cdot \mathrm{pk}^r$.    Output $m$.
    Output $(\mathrm{pk}, \mathrm{sk})$.    Output $(c_1, c_2)$.
```

This is OW-CPA secure assuming the Diffie–Hellman problem for the group G is hard. Hence, the KEM derived from ElGamal by Dent's construction, see Appendix A, is IND-CCA2 secure.

We now derive an m-KEM from ElGamal by letting the key generation function be as in standard ElGamal. We then define encapsulation and decapsulation via

```
$\mathcal{E}_{mKEM}(\{\mathrm{pk}_1, \dots, \mathrm{pk}_n\})$:
    $m \xleftarrow{R} G$.
    $r \leftarrow H(m)$.
    $c_0 \leftarrow g^r$.
    $c_i \leftarrow m \cdot \mathrm{pk}_i^r$ for $i = 1, \dots, n$.
    $K \leftarrow KDF(m)$.
    $C \leftarrow (c_0, c_1, \dots, c_n)$.
    Output $(K, C)$.

$\mathcal{D}_{mKEM}(C, \mathrm{sk}_i)$:
    Parse $C$ as $(c_0, c_1, \dots, c_n)$.
    $m \leftarrow c_i / c_0^{\mathrm{sk}_i}$.
    $r \leftarrow H(m)$.
    If $c_0 \neq g^r$ the output $\perp$ and halt.
    $K \leftarrow KDF(m)$.
    Output $K$.
```

Later we shall prove

Theorem 2. *If n grows as a polynomial function of the security parameter k and the Diffie–Hellman problem in G is hard then, in the random oracle the above m-KEM is secure, for n users, in the sense of (m, n)-IND-CCA2 for m-KEMs for all $m \leq n$.*

6 An ID-m-KEM

In this section we present an ID-m-KEM which, when combined with a suitable DEM, is more efficient at encrypting a large message to a large number of identities than repeated application of the Boneh–Franklin ID-IND-CCA2 [3] scheme to the DEMs session key.

However, before presenting the ID-m-KEM in full we first present an ID-based encryption scheme which encrypts to n-participants at once. We call such a scheme an m-ID encryption scheme. The following scheme will form the basis of our ID-m-KEM and is derived from the more general system described in [8], where the underlying ideas were used in the context of "cryptographic workflow".

Our scheme is based on the Boneh–Franklin system. We will assume we have a cyclic subgroup G_1 of prime order $q \approx 2^k$ of an elliptic curve and a subgroup G_2 of a finite field such that there is a computable non-degenerate bilinear pairing

$$t : G_1 \times G_1 \longrightarrow G_2.$$

We write the group operation additively in G_1 and multiplicatively in G_2, and let P denote a generator of G_1. We let the domain parameters be given by

$$\mathbb{D} = (q, P, G_1, G_2, g, t).$$

We shall assume that in this setting the Bilinear Diffie–Hellman problem (BDH) is hard. This means that given P, aP, bP, cP, for $P \in G_1$ and $a, b, c \in \mathbb{F}_q^*$, it is hard to compute $t(P,P)^{abc} \in G_2$.

We need to define some cryptographic hash functions

$$H_1 : \{0,1\}^* \longrightarrow G_1,$$
$$H_2 : G_2^* \longrightarrow \{0,1\}^l.$$

To every identity ID we associate the point Q_{ID} via

$$Q_{\mathrm{ID}} = H_1(\mathrm{ID}).$$

The point Q_{ID} is the public key associated with the identity ID. The m-ID encryption scheme is then given by the following set of algorithms.

- $\mathcal{G}_{m-ID}(\mathbb{D})$ the probabilistic key generation algorithm for the trust authority.
- $\mathcal{X}_{m-ID}(\mathrm{ID}, \mathrm{sk})$ is the trust authorities key extraction algorithm.
- $\mathcal{E}_{m-ID}(m, \mathcal{I}, R; r)$ which is a probabilistic encryption algorithm which takes as a message m (of l bits in length), n-identities $\mathcal{I} = \{\mathrm{ID}_1, \ldots, \mathrm{ID}_n\}$, the trust authority public key R and the randomness $r \in \mathbb{F}_q^*$.
- $\mathcal{D}_{m-ID}(c, S_{\mathrm{ID}_i}, R)$ which is a decryption algorithm. This takes as input a ciphertext c, a private key S_{ID_i} and the trust authorities key R and outputs the message m.

```
$\mathcal{G}_{m-ID}(\mathbb{D})$:
    $\mathrm{sk} \xleftarrow{R} \mathbb{F}_q^*$.
    $\mathrm{pk} \leftarrow R \leftarrow \mathrm{sk}P$.
    Output $(\mathrm{pk}, \mathrm{sk})$.
```

```
$\mathcal{X}_{m-ID}(\mathrm{ID}, \mathrm{sk})$:
    $S_{\mathrm{ID}} \leftarrow \mathrm{sk}Q_{\mathrm{ID}}$.
    Output $S_{\mathrm{ID}}$.
```

```
$\mathcal{E}_{m-ID}(m, \mathcal{I}, R; r)$:
    $U \leftarrow rP$.
    $T_i \leftarrow Q_{\mathrm{ID}_{i+1}} - Q_{\mathrm{ID}_1}$ for $i < n$.
    $U_i \leftarrow rT_i$ for $i < n$.
    $K \leftarrow H_2(t(R, rQ_{\mathrm{ID}_1}))$.
    $V \leftarrow m \oplus K$.
    Output $c = (U, U_1, \ldots, U_{n-1}, V)$.
```

```
$\mathcal{D}_{m-ID}(c, S_{\mathrm{ID}_i}, R)$:
    $t_1 \leftarrow t(U, S_{\mathrm{ID}_i})$.
    If $i = 1$ then $K \leftarrow H_2(t_1)$.
    If $i \neq 1$ then
        $t_2 \leftarrow t(R, -U_{i-1})$.
        $K \leftarrow H_2(t_1 \cdot t_2)$.
    $m \leftarrow V \oplus K$.
    Output $m$.
```

Note in the case where $n = 1$ this becomes the scheme BasicIdent of [3]. We define OW-security for an m-ID scheme encryption scheme in the obvious way. We say such a scheme is n-OW secure if in the second stage the adversary is passed a ciphertext encrypted to at most m identities. Clearly the above scheme is 1-OW secure since BasicIdent is secure by Theorem 4.1 of [3] assuming the BDH problem is hard. We conjecture that the above scheme is n-OW secure for all n as n grows as a polynomial function of the security parameter.

Using the above m-ID encryption scheme we define our ID-m-KEM as follows: The trust authority key generation $\mathcal{G}_{ID-mKEM}$ and key extraction algorithms $\mathcal{X}_{ID-mKEM}$ are as $\mathcal{G}$ and $\mathcal{X}$ above. The encapsulation mechanism then makes use of the analogue of Dent's construction, see Appendix A.

```
$\mathcal{E}_{ID-mKEM}(\mathcal{I}, R)$:
    $m \overset{R}{\leftarrow} \mathcal{M}$.
    $r \leftarrow H(m)$.
    $C \leftarrow \mathcal{E}_{m-ID}(m, \mathcal{I}, R; r)$.
    $K \leftarrow KDF(m)$.
    Output $(K, C)$.

$\mathcal{D}_{ID-mKEM}(C, S_{\mathrm{ID}}, \mathcal{I})$:
    $m \leftarrow \mathcal{D}_{m-ID}(C, S_{\mathrm{ID}_i}, R)$.
    $r \leftarrow H(m)$.
    Parse $C$ as $(U, U_1, \ldots, U_{n-1}, V)$
    If $U \neq rP$ then output $\perp$ and halt.
    $K \leftarrow KDF(m)$.
    Output $K$.
```

Later we shall prove the following theorem

Theorem 3. *If n grows as a polynomial function of the security parameter k then, in the random oracle model the above ID-m-KEM is secure in the sense of restricted $(1, n)$-ID-IND-CCA2 for ID-m-KEMs, assuming the BDH assumption holds.*

We conjecture that the above ID-m-KEM is in fact (m, n)-ID-IND-CCA2 secure for all $m \leq n$, but have been unable to prove this.

7 Efficiency Comparison

We first compare our ElGamal based m-KEM for n users against naive concatenation of n ElGamal ciphertexts together. We let $EG(n)$ denote a IND-CCA2 version of ElGamal (such as EC-IES/DH-IES [1]) applied n-times to encrypt a session key. We let $EG_{KEM}(n)$ denote the ElGamal based m-KEM described in Section 5.2 applied to n public keys. We compare the number of group exponentiations performed in the following table:

	$EG_{KEM}(n)$	$EG(n)$
Encapsulation	$n+1$	$2n+2$
Decapsulation	2	1

Hence we see that our method is more efficient than simply concatenating n-ElGamal ciphertexts together. In addition our method only requires the transmission of $n+1$ group elements as opposed to $2n$ group elements for the naive method.

We now compare our construction of an ID-m-KEM to that of applying the ID-IND-CCA2 Boneh–Franklin ID encryption function a number of times as to encrypt a session key. We suppose we wish to transmit a large message to n recipients, we ignore the number of additions in G_1 and multiplications in G_2 since these are negligible in comparison to the number of pairing computations (P) or group exponentiations (E).

	ID-m-KEM	ID-IND-CCA2
Encapsulation	$1P + (n+1)E$	$nP + 2nE$
Decapsulation	$2P + 1E$	$1P + 1E$

Hence, one can see that our methodology is much more efficient from the sender's perspective.

8 Additional ID-based Techniques

In this section we describe additional techniques which allow various improvements in the identity based setting.

8.1 Point Compression

In pairing based systems we need to transmit various elliptic curve points, e.g. the trust authorities key R or a component of the ciphertext U or U_i. Each point P consists of an x and y coordinate, but each x coordinate can only give rise to at most two y coordinates. A standard way of reducing bandwidth in elliptic curve based systems is to transmit x plus a single bit stating which value of y to take.

However, in pairing based systems we have another possibility. We can simply transmit the x and leave the reciever to decide for themselves which value of y to take. Due to the elliptic curve group law the recievers and senders value of P will differ by at most a change in sign,

$$P_S = \pm P_R.$$

At some point in a pairing based protocol one evaluates a pairing at two points, if one (or both) of the points are only determined up to sign then the evaluation will be determined up to inverses, i.e.

$$s = t(\pm P, \pm Q) = t(P, Q)^{\pm 1}.$$

This ambiguity can be removed by taking

$$z = s + \frac{1}{s}.$$

Hence in the above ID-based protocols one would apply H_2 to z rather than s. Such a remark applies to both the above m-ID scheme and ID-m-KEM, it also applies to the Boneh–Franklin scheme [3] and it can be adapted to work with the various signature schemes (both traditional and ID based) which are based on pairings that can be found in the literature.

8.2 Intersecting Domains

In the above ID-m-KEMs and in the Boneh–Franklin ID-based encryption scheme one uses the trust authorities public key to define a certain domain of trust. Recall this public key is given by

$$R = \mathrm{sk}P,$$

where sk is the secret key of the trust authority and P is some fixed public point. The trust authority is responsible for providing users with their secret keys, and as such it vouches for the identity of each user.

In various situations one can imagine sending an email where the sender does not wish to soley relie on the proceedures of the recievers trust authority with public key $R_1 = s_1P$. However, the sender may trust another trust authority with public key $R_2 = s_2P$. The sender can obtain the effect of intersecting the two trust domains, if they have the same domain parameters, by simply adding the two public keys together to obtain a public key for a virtual trust authority

$$R = R_1 + R_2 = (s_1 + s_2)P.$$

The associated user secret keys for the new virtual domain are obtained by addings the two secret keys for two original domains, i.e.

$$S_{\mathrm{ID}} = S_{\mathrm{ID}}^{(1)} + S_{\mathrm{ID}}^{(2)} = (s_1 + s_2)Q_{\mathrm{ID}}.$$

Clearly such a technique not only applies to encryption techniques but can also be applied to identity based signature techniques based on pairings, such as that of Hess. In such a signature scenario it helps mitigate the problems associated with key-escrow and non-repudiation.

References

1. M. Abdalla, M. Bellare and P. Rogaway. DHAES : An encryption scheme based on the Diffie–Hellman problem. Preprint, 1999.
2. M. Bellare, A. Boldyreva and S. Micali. Public-key encryption in the multi-user setting : Security proofs and improvements. In *Advances in Cryptology – EuroCrypt 2000*, Springer-Verlag LNCS 1807, 259–274, 2000.
3. D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO 2001*, Springer-Verlag LNCS 2139, 213–229, 2001.
4. C. Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding*, Springer-Verlag LNCS 2260, 360–363, 2001.
5. A.W. Dent. A designer's guide to KEMs. To appear *Coding and Cryptography*, Springer-Verlag LNCS XXXX, XXXX–XXXX, 2003.
6. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen-ciphertext attack. Preprint, 2002.
7. V. Shoup. A proposal for the ISO standard for public-key encryption (version 2.0). Preprint, 2001.
8. N.P. Smart. Access control using pairing based cryptography. In *Topics in Cryptology – CT-RSA 2003*, Springer-Verlag LNCS 2612, 111–121, 2003

A Review of Dent's Construction

In this section we recap on some prior work on classical KEM's, in particular a construction of a secure KEM given a public key encryption algorithm which is secure in the sense of OW-CPA [5].

We first turn our attention to probabilistic public key encryption schemes. We formally define these as a triple of algorithms:

- $\mathcal{G}(\mathbb{D})$ which is a probabilistic key generation algorithm. On input of $\mathbb{D}$, the domain parameters, this algorithm outputs a public/private key pair (pk, sk).
- $\mathcal{E}(m, \mathrm{pk})$ which is a probabilistic public key encryption algorithm. On input of a public key pk and a message $m \in \mathcal{M}$ this algorithm output a ciphertext c, it makes use of a random value drawn from a space $\mathcal{R}$.
- $\mathcal{D}(c, \mathrm{sk})$ which is a decryption algorithm. This takes as input a ciphertext c and a private key sk and outputs the associated message m or a special symbol $\perp$ representing the case where c is an invalid ciphertext with respect to the private key sk.

For such a scheme to be useful we require that it is sound in the following sense,

$$\Pr\Big((\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{G}(\mathbb{D}), m \overset{R}{\leftarrow} \mathcal{M}, c \leftarrow \mathcal{E}(m, \mathrm{pk}) : m = \mathcal{D}(c, \mathrm{sk})\Big) = 1.$$

We also require that the scheme is truly probabilistic in that the proportion of values of r, used as input into $\mathcal{E}(m, \mathrm{pk}; r)$, that encrypt a given message to a given ciphertext is negligible as a function of the security parameter.

We shall require the security notion of OW-CPA for public key schemes, which we recap on now. We assume an adversary $\mathcal{A}$ which takes a challenge ciphertext c^* and a public key and is asked to produce the associated plaintext. The scheme said to be OW-CPA secure if no adversary exists which wins the following game with probability greater than a negligible function of the security parameter k.

```
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{G}(\mathbb{D})$.
$m \overset{R}{\leftarrow} \mathcal{M}$.
$c^* \leftarrow \mathcal{E}(m, \mathrm{pk})$.
$m' \leftarrow \mathcal{A}(\mathrm{pk}, c^*)$.
Output whether $m = m'$.
```

The adversary is not given access to any decryption oracles, but is clearly allowed to encrypt arbitrary messages of its choice since it has access to pk.

Dent [5] proposes a KEM, derived from a OW-CPA probabilistic public key algorithm $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, a hash function H with codomain $\mathcal{R}$ (the space of randomness used by algorithm $\mathcal{E}$) and a key derivation function KDF with domain $\mathcal{M}$. Dent's scheme is described as follows:

$\mathcal{G}_{KEM}(\mathbb{D})$: $\mathcal{G}_{KEM} = \mathcal{G}$.

```
$\mathcal{E}_{KEM}(\mathrm{pk})$:
    $m \xleftarrow{R} \mathcal{M}$.
    $r \leftarrow H(m)$.
    $C \leftarrow \mathcal{E}(m, \mathrm{pk}; r)$.
    $K \leftarrow KDF(m)$.
    Output $(K, C)$.
```

```
$\mathcal{D}_{KEM}(C, \mathrm{sk})$:
    $m \leftarrow \mathcal{D}(C, \mathrm{sk})$.
    If $m = \perp$ then output $\perp$ and halt.
    $r \leftarrow H(m)$.
    Check that $C = \mathcal{E}(m, \mathrm{pk}; r)$,
        if not output $\perp$ and halt.
    $K \leftarrow KDF(m)$.
    Output $K$.
```

One then has the following result, when one models the functions H and KDF as random oracles,

Theorem 4 (Dent [5]). *If $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is a OW-CPA probabilistic public key encryption algorithm then in the random oracle the KEM $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ derived from $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is secure in the sense of IND-CCA2.*

B Proof of Theorem 1

Since $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is IND-CCA2 secure as a public key encryption algorithm it is secure in the multi-user setting described in [2].

We recap on the security model from [2]. The adversary is given n public keys $\mathrm{pk}_1, \ldots, \mathrm{pk}_n$ and is given access to a left-right oracle $\mathcal{O}_{LR}$ which on input of $\{\{m_0, m_1\}, \mathrm{pk}_i\}$ will output the encryption of m_b under pk_i for some fixed hidden bit b. The adversary is given access to a decryption oracle $\mathcal{O}_D$ for all the public keys pk_i, subject to the constraint it is not allowed to ask for the decryption of the result of a call to the left-right oracle $\mathcal{O}_{LR}$.

We assume an adversary $\mathcal{A}$ against the mKEM $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ and show how one can use this to produce an adversary $\mathcal{B}$ against $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ in the above multi-user setting. Thus we will derive a contradiction.

Algorithm $\mathcal{B}$ takes as input the n public keys $\mathcal{P}' = \{\mathrm{pk}_1, \ldots, \mathrm{pk}_n\}$. These are then passed into algorithm $\mathcal{A}^1$. We answer the decapsulation oracle queries of $\mathcal{A}^1$ using the decryption provided to $\mathcal{B}$ in an obvious way, i.e. on input of $C = (c_1, \ldots, c_n)$ with respect to some public key pk_i we execute

```
$m \leftarrow \mathcal{O}_D(c_i, \mathrm{pk}_i)$.
If $m = \perp$ then output $\perp$ and halt.
$K \leftarrow KDF(m)$.
Output $K$.
```

Algorithm $\mathcal{A}^1$ eventually will terminate and will return a list of public keys

$$\mathcal{P} = \{\mathrm{pk}_{i_1}, \ldots, \mathrm{pk}_{i_m}\} \subset \mathcal{P}',$$

and a state s.

Algorithm $\mathcal{B}$ then computes two random messages $m_0, m_1 \in \mathcal{M}$ and computes $K_0 = KDF(m_0)$ and $K_1 = KDF(m_1)$. Then using the left-right oracles it computes

$$C^* = (c_{i_1}, \ldots, c_{i_m})$$

where

$$c_{i_j} = \mathcal{O}_{LR}(\{m_0, m_1\}, \mathrm{pk}_{i_j}).$$

One then executes $\mathcal{A}^2(C^*, \{K_0, K_1\}, s)$. The decapsulation oracle queries of $\mathcal{A}^2$ are answered as above on noting that any oracle query allowed in the game being played by $\mathcal{A}^2$ will be able to be answered by the oracle provided to $\mathcal{B}$.

Finally $\mathcal{A}^2$ will respond with its guess b' as to the hidden bit b, we let this bit b' be the output of $\mathcal{B}$. If $\mathcal{A}^2$ answers correctly then algorithm $\mathcal{B}$ will also answer correctly.

C Proof of Theorem 2

We let $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ denote the ordinary KEM derived from the ElGamal system via Dent's transform for OW-CPA probabilistic public key algorithms. It is known, by Theorem 4, that in the random oracle model the scheme $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ is secure in the IND-CCA2 sense assuming the Diffie–Hellman problem is hard.

We let $(\mathcal{G}_{mKEM}, \mathcal{E}_{mKEM}, \mathcal{D}_{mKEM})$ denote our m-KEM. We shall assume we have an IND-CCA2 adversary $\mathcal{A}$ against this scheme in the random oracle model which works against n public keys. We shall show how to use $\mathcal{A}$ to create an adversary $\mathcal{B}$ against $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$. Since such an adversary is assumed, in the random oracle model, not to exist we can then conclude that $\mathcal{A}$ could not exist either.

We first describe algorithm $\mathcal{B}^1(\mathrm{pk})$. Let $\mathrm{pk}_1 = \mathrm{pk}$ denote the public key input into algorithm $\mathcal{B}^1$. We first generate some extra public keys via, $k_i \xleftarrow{R} \mathbb{F}_q^*$ and $\mathrm{pk}_i = \mathrm{pk} \cdot g^{k_i}$, for $i = 2, 3, \ldots, n$. We now pass the set $\mathcal{P}' = \{\mathrm{pk}_1, \ldots, \mathrm{pk}_n\}$ into $\mathcal{A}^1$. We then obtain a subset $\mathcal{P} \subset \mathcal{P}'$ and a state s'. We shall discuss how to answer all decapsulation oracle queries of $\mathcal{A}^1$ later. We let s denote the state

$$s = \{(k_2, \mathrm{pk}_2), \ldots, (k_n, \mathrm{pk}_n), \mathcal{P}, s'\}$$

and return s as the output of $\mathcal{B}^1(\mathrm{pk})$.

Algorithm $\mathcal{B}^2$ takes as input two keys K_0 and K_1, the state information s and an encapsulation C^* of one of the keys $K_b \in \{K_0, K_1\}$ from the algorithm $\mathcal{E}_{KEM}$ with respect to the public key pk. We first need to create a valid encapsulation C_m^* of the key K_b with respect to the algorithm $\mathcal{E}_{mKEM}$ and the set of keys $\mathcal{P} = \{\mathrm{pk}_{i_1}, \ldots, \mathrm{pk}_{i_m}\}$. We have

$$C^* = (c_0^*, c_1^*) = (g^{\mathfrak{r}}, \mathfrak{m} \cdot \mathrm{pk}^{\mathfrak{r}}),$$

with $K_b = KDF(\mathfrak{m})$ and $\mathfrak{r} = H(\mathfrak{m})$, whereas

$$C_m^* = (c_0^*, c_1^*, \ldots, c_m^*) = (g^{\mathfrak{r}}, \mathfrak{m} \cdot \mathrm{pk}_{i_1}^{\mathfrak{r}}, \ldots, \mathfrak{m} \cdot \mathrm{pk}_{i_m}^{\mathfrak{r}}).$$

Hence, we set $c_0^* = c_0^*$ and for $j = 1, \ldots, m$

$$\begin{aligned} c_j^* &= c_1^* \cdot c_0^{*k_{i_j}} \\ &= \mathrm{m} \cdot \mathrm{pk}^r \cdot g^{rk_{i_j}} = \mathrm{m} \cdot \left(\mathrm{pk} \cdot g^{k_{i_j}}\right)^r \\ &= \mathrm{m} \cdot \mathrm{pk}_{i_j}^r, \end{aligned}$$

where we let $k_1 = 0$.

Having constructed C_m^* we can now pass $C_m^*, \{K_0, K_1\}$ and s' to algorithm $\mathcal{A}^2$. If $\mathcal{A}$ is a successful adversary against the mKEM then we will obtain with non-negligible probability a bit b' such that $K_b = K_{b'}$. Hence, by returning this bit b' as our output from the algorithm $\mathcal{B}^2$ we obtain an algorithm $\mathcal{B}$ which with non-negligible probability will break the security of the $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ in the IND-CCA2 sense.

All that remains is to show how to answer the decapsulation oracle queries of algorithm $\mathcal{A}$. Recall we have a decapsulation oracle $\mathcal{O}_{KEM}$ for the scheme $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ which will respond with respect to the key pk on all requests C except one is not allow to query it with $C = C^*$. The decapsulation queries of $\mathcal{A}$ must be answered correctly unless the query C_m corresponds to the key K_b.

Suppose we are given the, possibly invalid, encapsulation

$$C_m = (c_0, c_1, \ldots, c_m) = (g^r, m \cdot \mathrm{pk}_{i_1}^{r_1}, \ldots, m \cdot \mathrm{pk}_{i_m}^{r_m})$$

and we are asked to decapsulate it with respect to the public key pk_{i_j}. This should result in the key $K = KDF(m)$ if and only if $r = H(m)$ and $r_j = r$.

We first form the encapsulation $(\mathrm{c}_0, \mathrm{c}_1)$ with respect to the scheme KEM, via setting $\mathrm{c}_0 = c_0$ and

$$\begin{aligned} \mathrm{c}_1 &= c_{i_j} \cdot c_0^{-k_{i_j}} \\ &= m \cdot \mathrm{pk}_{i_j}^{r_j} \cdot g^{-rk_{i_j}} \\ &= m \cdot \mathrm{pk}^{r_j} \cdot g^{(r_j - r)k_{i_j}} \\ &= m \cdot \mathrm{pk}^r \text{ if } r_j = r. \end{aligned}$$

Note since we are not allowed to query $\mathcal{A}$'s decapsulation oracle with any encapsulation which corresponds to K_b we must have $m \neq \mathrm{m}$.

The oracle $\mathcal{O}$ will not respond if $\mathrm{c}_0 = \mathrm{c}_0^*$ and $\mathrm{c}_1 = \mathrm{c}_1^*$. Such a situation would mean that $\mathcal{O}$ returns K_b, i.e. the encapsulation C_m is an encapsulation of K_b with respect to pk_{i_j}, and such a query is invalid under the security model for mKEMs.

We can, therefore, assume that either $\mathrm{c}_0 \neq \mathrm{c}_0^*$ or $\mathrm{c}_1 \neq \mathrm{c}_1^*$. In either case the oracle $\mathcal{O}$ will compute

$$m' = m \cdot \mathrm{pk}^{r_j} \cdot g^{(r_j - r)k_{i_j}} \cdot \mathrm{c}_0^{-\mathrm{sk}} = m \cdot \left(\mathrm{pk} g^{k_{i_j}}\right)^{r_j - r}.$$

For the oracle $\mathcal{O}$ to return $K = KDF(m')$ we must have $r = H(m')$. In which case (c_0, c_1) is a (possibly badly formed) ciphertext for the KEM which never-the-less passes the validity check and C_m is a (possibly badly formed) ciphertext for the mKEM which also passes the validity check of the mKEM. Hence, $\mathcal{A}$s oracle should return K, unless $K = K_b$ in which case we have found a collision in KDF since

$$KDF(m) = KDF(m').$$

Such a collision will only occur with negligible probability since KDF is modelled as a random oracle.

D Proof of Theorem 3

First consider the scheme BasicPub from [3], which we recap on. The key generation algorithm is given by

$Q_{\mathrm{ID}} \xleftarrow{R} \langle P \rangle.$
$\mathrm{sk} \xleftarrow{R} \mathbb{F}_q^*.$
$S_{\mathrm{ID}} \leftarrow \mathrm{sk} Q_{\mathrm{ID}}.$
$P_{pub} \xleftarrow{R} \mathrm{sk} P.$

The public key is given by $\mathrm{pk} = \{P_{pub}, Q_{\mathrm{ID}}\}$ and the secret key is given by S_{ID}. Encryption $(U, V) \leftarrow \mathcal{E}(m, \mathrm{pk})$ is given by

$r \xleftarrow{R} \mathbb{F}_q^*.$
$U \leftarrow rP.$
$V \leftarrow m \oplus H_2(t(P_{pub}, rQ_{\mathrm{ID}})).$

Decryption in BasicPub is given by

$$m \leftarrow V \oplus H_2(t(U, S_{\mathrm{ID}})).$$

Assuming the BDH assumption holds for the set of domain parameters, Boneh and Franklin show that this public key scheme is secure in the OW-CPA model. Notice, that BasicPub is the same as BasicIdent except that the identity is now a random identity chosen at the time of generation of the public key.

We let $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ denote the KEM derived from BasicPub using Dent's method. By Dent's Theorem, Theorem 4, this KEM is secure in the sense of IND-CCA2.

We let $\mathcal{A}$ denote an adversary, in the sense of $(1, n)$-ID-IND-CCA2, against our $ID - m - KEM$ given by $(\mathcal{G}_{ID-mKEM}, \mathcal{E}_{ID-mKEM}, \mathcal{D}_{ID-mKEM})$. We will construct an adversary $\mathcal{B}$ which breaks $(\mathcal{G}_{KEM}, \mathcal{E}_{KEM}, \mathcal{D}_{KEM})$ in the sense of IND-CCA2, this will give us our necessary contradiction. In our proof we shall model the function H_2 as a random oracle.

Algorithm $\mathcal{B}^1$ takes as input a public key $\mathrm{pk} = \{P_{pub}, Q_{\mathrm{ID}}\}$. We let R, the trust authorities public key, be given by P_{pub}. If algorithm $\mathcal{A}$ requires the extraction of the secret key with respect to an identity ID_i, or evaluation of H_1 at ID_i then we fix the oracle calls of H_1 so that

$Q_{\mathrm{ID}_i} = r_i P = H_1(\mathrm{ID}_i)$.
$S_{\mathrm{ID}_i} = r_i R$.

Notice that for all i we have $S_{\mathrm{ID}_i} = \mathrm{sk}Q_{\mathrm{ID}_i}$, hence the S_{ID_i} are valid secret keys for the identities ID_i.

We generate a random identity ID and define H_2 so that $Q_{\mathrm{ID}} = H_1(\mathrm{ID})$. We also generate a set of n identities

$$\mathcal{P}' = \{\mathrm{ID}_1, \ldots, \mathrm{ID}_n\}$$

in such a way that for some $j \in \{1, \ldots, n\}$

$\mathrm{ID}_j = \mathrm{ID}$.
$Q_{\mathrm{ID}_i} = r_j P = H_1(\mathrm{ID}_j)$ for $i \neq j$.
$S_{\mathrm{ID}_i} = r_i R$ for $i \neq j$.

We pass $\mathcal{P}'$ to the algorithm $\mathcal{A}^1$, answering it's decapsulation queries with respect to ID_i for $i = j$ via the decapsulation oracle of $\mathcal{B}$ and the other decapsulation queries using the associated secret key. At the end of algorithm $\mathcal{A}^1$, since $\mathcal{A}$ is a $(1, n)$-ID-IND-CCA2 adversary, we obtain a single identity $\mathrm{ID}' \in \mathcal{P}'$ plus the internal state of $\mathcal{A}^1$.

If $\mathrm{ID}' \neq \mathrm{ID}$ then algorithm $\mathcal{B}^1$ halts and restarts. Since $j \in \{1, \ldots, n\}$ is chosen out of the view of algorithm $\mathcal{A}^1$ and the identities are chosen randomly, the probability that one obtains $\mathrm{ID}' = \mathrm{ID}$ from algorithm $\mathcal{A}^1$ is at least $1/n$. Eventually, after at most polynomially many trials, algorithm $\mathcal{B}^1$ will then terminate by outputing its complete internal state s.

Algorithm $\mathcal{B}^2$ is defined as follows. It is passed the internal state s of $\mathcal{B}^1$, two keys K_0 and K_1 plus an encapsulation C^* of one of these keys K_b under $\mathcal{E}_{KEM}$, for some hidden bit b. The encapsulation is defined by

$$C^* = (U^*, V^*)$$

where, for some random nonce r,

$$U^* = rP, \quad V^* = K_b \oplus H_2(t(P_{pub}, rQ_{\mathrm{ID}})).$$

Algorithm $\mathcal{B}^2$ then takes this as the challenge encapsulation on the identity $\mathcal{P} = \{\mathrm{ID}\}$ under $\mathcal{E}_{m-ID-KEM}$. Algorithm $\mathcal{A}^2$ is then passed the internal state of $\mathcal{A}^1$, the identity $\mathcal{P} = \{\mathrm{ID}\}$, the keys $\{K_0, K_1\}$ and the challenge C^*. Algorithm $\mathcal{A}^2$ returns with its guess as to the hidden bit b. This bit is returned as the output of $\mathcal{B}^2$. Clearly if $\mathcal{A}^2$ is successful then so is $\mathcal{B}^2$.

The decapsulation and extraction queries of $\mathcal{A}^2$ are handled as for $\mathcal{A}^1$. It is gauranteed by the security model for $\mathcal{A}^2$ that at no point will a disallowed oracle query to the oracle provided by $\mathcal{B}^2$ will be requested.

Document made available under the Patent Cooperation Treaty (PCT)

International application number:	PCT/EP04/012226
International filing date:	28 October 2004 (28.10.2004)

Document type:	Certified copy of priority document	
Document details:	Country/Office:	GB
	Number:	0325225.1
	Filing date:	29 October 2003 (29.10.2003)

Date of receipt at the International Bureau: 24 January 2005 (24.01.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)